

AS LINGUAGENS DE PROGRAMAÇÃO PARA ARTES – METODOLOGIAS DE ENSINO-APRENDIZAGEM ADAPTADAS

Madalena Ribeiro

Instituto Politécnico de Castelo Branco -Escola Superior de Artes Aplicadas

RESUMO: Atualmente, as linguagens de programação são usadas por pessoas com formação em domínios cada vez mais diversos e distintos do habitual, tendo já chegado ao domínio das artes. A linguagem Processing, que tem um cariz visual e que possibilita o desenvolvimento de aplicações/performances com elementos visuais e sonoros, foi mesmo projetada para ser usada por artistas. Contudo, a preparação prévia e a sensibilidade do público-alvo desta linguagem, enquanto aprendizes, é bastante distinta da preparação dos alunos comuns de linguagem de programação. Por esse motivo, é aqui proposto um conjunto de metodologias de ensino/aprendizagem adaptadas, no sentido de criar entusiasmo na aprendizagem e atingir um êxito efetivo.

Palavras Chave: Linguagens de programação; Artes; Ensino-Aprendizagem

ABSTRACT: Today, the programming languages are used by people with background from areas increasingly distinct and different from the usual, having come to the artistic field. The Processing language, which has a visual nature allowing the development of applications/ performances with visual elements and sounds, was specially designed to be used for artists. Nevertheless, the preparation prior and the sensitivity of the target audience of this language has, while apprentices, is quite different from the preparation of the programming language standard learners. For this reason, it is proposed here a set of teaching/learning methodologies, in order to create enthusiasm in the learning process and to get an effective success.

Keywords: Programming Languages, Arts, Teaching and Learning

Introdução

As unidades curriculares de iniciação às linguagens de programação fazem parte de um vasto conjunto de cursos de licenciatura, especialmente das áreas das engenharias ou afins, com um enfoque especial em determinados domínios, nomeadamente na informática, eletrónica, eletrotécnica, física, matemática e física.

Os alunos que integram estas licenciaturas acedem a este grau de ensino com uma preparação prévia no ensino secundário, que lhes permite uma compreensão de determinadas estruturas sintáticas e semânticas. Porém, atualmente, estas unidades curriculares são também já lecionadas a cursos do domínio das Artes e do Design. Uma vez que as aprendizagens prévias desses alunos é completamente distinta, apresenta-se como fundamental que as metodologias usadas sejam também elas diferenciadas.

Neste contexto, foi criada uma forma adaptada de lecionar as unidades curriculares de Linguagens de Programação I e Linguagens de Programação II, para os alunos da licenciatura em Música – variante de Música Eletrónica e Produção Musical da Escola Superior de Artes Aplicadas (ESART) do Instituto Politécnico de Castelo Branco (IPCB). Na sua essência, as adaptações residem na adaptação de uma linguagem de programação de carácter visual, assim como um conjunto de estratégias metodológicas, para

possibilitar um ambiente de ensino-aprendizagem inspirador e o sucesso da aprendizagem.

Linguagens de Programação para as Artes

A congregação e a partilha de experiências e de saberes entre as diversas áreas do conhecimento, tem vindo a criar condições para a obtenção de resultados inesperados e a revelar resultados que há algumas décadas se pensariam impossíveis. Com o desenvolvimento e democratização dos recursos computacionais, as linguagens de programação passaram a ser usadas como ferramenta comum, no desenvolvimento de soluções adequadas às mais diversas áreas da ciência. Muitos dos profissionais dessas áreas passaram a integrar, nas suas práticas diárias correntes, o desenvolvimento de programas para resolução de problemas concretos.

Contudo, numa primeira perspetiva, o sucesso na introdução de linguagens de programação em determinados domínios do saber nem sempre é imediato. Além do conhecimento sobre a sintaxe da linguagem, de conceitos sobre lógica e de construção de algoritmos, o desenvolvimento de software implica ainda uma capacidade de abstração própria, que se vai fortalecendo pelo estudo das disciplinas associadas às ciências. Por isso, a adoção do uso de linguagens de programação por pessoas ligadas ao domínio das artes, pode parecer, à primeira vista, uma ideia pouco viável.

Não obstante esta visão, a linguagem de marcação HTML, surgida em finais do século passado, veio abrir portas na utilização de linguagens de programação por parte de pessoas que não têm qualquer formação na área. O facto de ser especialmente acessível serviu de sustentação para que a maior parte das pessoas pudesse criar e publicar documentos na web, editando diretamente o código HTML necessário. A simplicidade da linguagem HTML constituiu um motor na sua popularização. Porém, a linguagem HTML não abarca o domínio de elementos basilares típicos das linguagens de programação, como sejam a lógica, as estruturas de controlo ou a algoritmia.

Entretanto, a intervenção das artes passou a assumir uma morfologia diversa e sucessivamente mais alargada. As barreiras entre áreas esbateram-se e os elementos artísticos de nova geração, como o vídeo, o som, a imagem e as imagens virtuais criadas artificialmente, ganharam preponderância. Por isso, o uso das tecnologias de informação nas artes tem vindo a ganhar importância e profundidade. Embora as aplicações WYSIWYG tenham vindo a oferecer um suporte precioso aos artistas, na combinação de conteúdos multimédia, estas não têm capacidade para dar resposta ao desenvolvimento de determinados efeitos/performances, devido à limitação da própria funcionalidade.

Neste contexto, e tendo em mente as pessoas cujo domínio de formação pertence às áreas das artes eletrónicas e do design visual, Casey Reas e Benjamin Fry, introduziram uma nova linguagem de programação, designada Processing, que possibilita uma exploração de ideias artísticas e a sua implementação, através de instruções computacionais, criando performances que envolvem imagem e som. Através dos seus livros (Reas et al., 2007) e (Reas et al., 2010), dão a conhecer exemplos de performances artísticas que é possível criar com recurso à linguagem, além de constituírem bons recursos de suporte para a aprendizagem da linguagem.

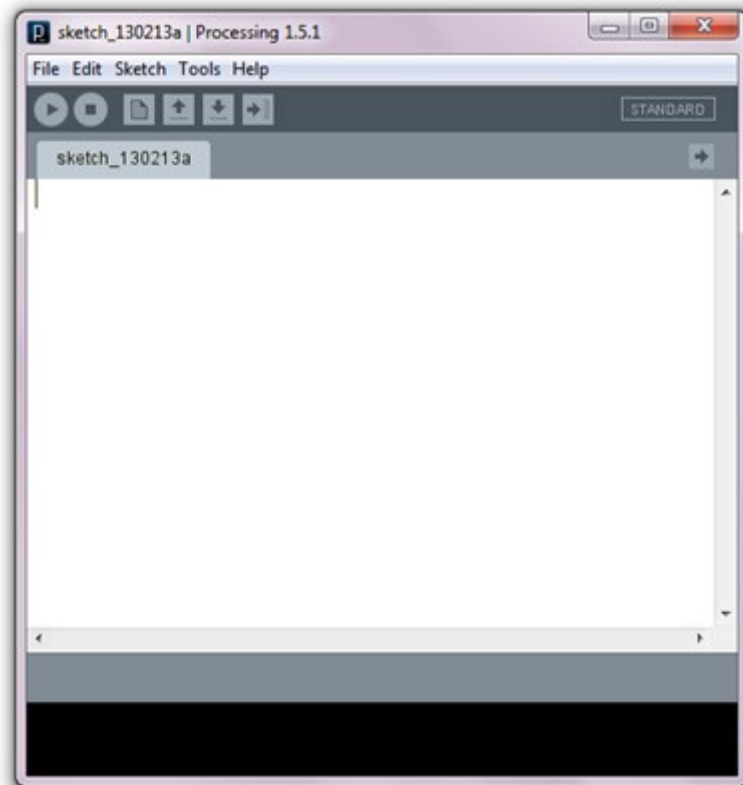


Figura 1 – Aspetto do ambiente de desenvolvimento do Processing.

Os autores criaram ainda um ambiente de desenvolvimento integrado (IDE) que possibilita o desenvolvimento de aplicações (programas) com recurso à linguagem Processing. Os ambientes de programação, habitualmente usados para o desenvolvimento de código, são complexos, pródigos em menus que dispõem de inúmeras funcionalidades, intimidando um utilizador de nível inicial. O ambiente do Processing é extremamente simples e intuitivo, levando o utilizador a sentir-se confortável e encorajado a experimentar as opções que tem à sua disposição.

O Ensino-Aprendizagem de Linguagens de programação

A aprendizagem de linguagens de programação é um processo complexo, uma vez que consiste num processo de resolução de problemas através da delineação de uma estratégia (pseudo-código), na escrita dessa estratégia através de um conjunto de instruções (algoritmo), que deve obedecer a uma linguagem específica (sintaxe) (Stair et al., 2003). Este processo pressupõe ainda alguns conhecimentos sobre lógica e o recurso à capacidade de abstração, de modo a poder modelar determinadas funções ou comportamentos.

Estas aptidões últimas (sobretudo a lógica) são saberes que se vão consolidando de uma forma gradual, pelo estudo das disciplinas associadas às ciências. Ao longo do tempo, durante o estudo das matérias desse campo (sobretudo na matemática), vão sendo interiorizados conceitos e raciocínios que, depois, na aprendizagem de programação de computadores, servem de suporte.

No entanto, o Processing destina-se a pessoas de áreas artísticas com uma preparação prévia particular. Por isso, as estratégias de ensino/aprendizagem no ensino de linguagens de programação a alunos de cursos artísticos, devem ser adaptadas, tendo em conta as limitações decorrentes da preparação que estes alunos têm. Essas metodologias devem ainda ter em conta a sensibilidade que este tipo de público-alvo revela para estímulos de carácter especialmente sensorial.

Esta tarefa fica simplificada pelas características apresentadas pelo ambiente de desenvolvimento e pela linguagem Processing. O ambiente de desenvolvimento é especialmente acessível; a sintaxe da linguagem de programação é particularmente inteligível; e a criação de resultados, através da implementação, é entusiasmante, pois é possível obter resultados interessantes de forma bastante simples e direta.

Problemas do Ensino de Programação

De uma forma geral, a aprendizagem da programação, ao nível da iniciação, evidencia uma série de dificuldades por parte dos alunos, esatando relacionados com o insucesso do processo de ensino-aprendizagem. São diversas as causas do insucesso, sendo referidas nos trabalhos de vários autores. Importa aqui referir os que se reconhecem como mais relevantes.

Jenkins vem apontar a questão da abstração como uma barreira, e refere ainda um outro fator muito importante que considera também um entrave: a sintaxe de uma linguagem, que é em realidade a linguagem percebida pelo computador, está interiorizada pelos profissionais da programação (e pelos professores) (Jenkins, 2002). No entanto, ainda é uma matéria em aprendizagem por parte dos alunos. Por isso, a dificuldade de comunicação com o computador (através da criação de instruções em código), de forma a obter o resultado esperado, pode revelar-se desesperante. Há a acrescer ainda a dificuldade de identificar os erros de sintaxe, a partir do que é expresso pelo compilador. A correção dos erros existentes na sintaxe não é tarefa simples, numa primeira fase de aprendizagem. Refere ainda que a aprendizagem inicial de linguagens de programação é uma multitarefa, pois, face a um problema, é necessário, em simultâneo, conceber a estratégia de solução e transpô-la para instruções, recorrendo à sintaxe.

Por fim, há ainda a referir que, quando se leciona a temática da Programação a iniciados, independentemente da linguagem proposta ou do paradigma adotado (sequencial ou orientado a objetos), parte-se de um determinado nível de abstração de conceitos, conteúdos, estruturas, e diagramas, que só depois são operacionalizados através de exercícios de aplicação. Entende-se que este nível de abstração é, de per si, o principal fator de insucesso no processo de ensino/aprendizagem. Esta ideia é corroborada pelos autores dos artigos (Gomes et al., 2008) e (Almeida et al., 2002), que defendem ainda que o nível de abstração leva ao desinteresse e à desmotivação.

Ensino-Aprendizagem de Processing – Metodologias Adaptadas

O ensino/aprendizagem inicial de uma linguagem de programação por parte de alunos que nunca tiveram contacto com linguagem de programação não é um processo simples, revestindo-se de inúmeras dificuldades, como foi descrito na secção anterior, e que se relaciona com a própria natureza da temática.

Não obstante a dificuldade da matéria, a sua aprendizagem não deverá constituir, por princípio, um assunto árduo e fastidioso. A adoção de metodologias adaptadas possibilita que a aprendizagem seja um processo criativo e desafiante, promovendo a motivação e a satisfação dos alunos e o consequente sucesso da aprendizagem. Nesse sentido, são apontadas nesta secção, algumas diretrizes metodológicas.

O Primeiro Programa

De uma forma geral, o primeiro programa que é proposto aos alunos, independentemente da linguagem de programação, é o programa que emite a mensagem “Hello world”. Genericamente, a instrução que faz sair a frase para o écran está escrita em inglês, e a mensagem também, o que leva a que o aluno nem perceba muito bem o que faz cada uma das palavras no programa. Além disso, o resultado não é especialmente entusiasmante, pelo que não desafia o iniciado a evoluir. Esta convicção é também exposta em (Fontes et al., 2008). Depois de ver este primeiro exemplo, o aluno fica à espera, de forma mais ou menos passiva, que lhe seja apresentado algo mais.

O primeiro programa em Processing deve começar por estabelecer o tamanho da janela de output, que constitui, desde logo, uma vantagem pela possibilidade que o aluno tem de controlar o ambiente que vai criar. Deve ser ainda inserida uma instrução que desenhe uma primitiva geométrica 2D. Este ponto de partida é essencial para que o aluno perceba que apenas com duas instruções pode obter um resultado bem visível.

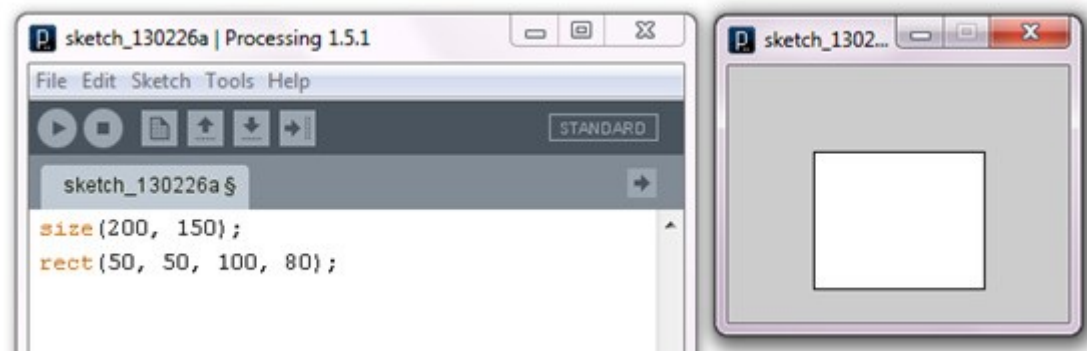


Figura 2 – Definição do tamanho da janela de output e desenho de um retângulo (código e resultado)

Ao ver a forma desenhada na janela de *output*, o aluno vai querer saber de imediato mais alguma coisa (*e.g.*, como alterar a cor de preenchimento, como alterar a cor ou a espessura de contorno, como criar um outra qualquer forma, como colocar o retângulo noutra posição ou como alterar as medidas do retângulo). Este entusiasmo que se pretende criar no aluno possibilita a procura autónoma de conhecimentos, desde o início da aprendizagem, promovendo uma atitude proactiva e uma aquisição de conhecimentos mais bem construída.

Ambiente Visual como Contexto

Tal como foi já referido, o Processing é uma linguagem essencialmente visual, com uma sintaxe bastante simplificada, que é especialmente visível em (Fry, 2007) ou em (Glassner, 2010). A possibilidade de criar efeitos visuais interessantes de forma quase imediata e simples é o principal atrativo desta linguagem.

As primitivas geométricas 2D (ponto, linha, elipse, retângulo, ...) são alguns dos elementos base, nesta linguagem de programação. Assim, a introdução de um exemplo com alguns (poucos) elementos gráficos deve constituir o ponto de partida para a abordagem do conceito de primitiva geométrica 2D. Estes elementos são adequados para serem usados de forma corrente, como utensílio, para abordar e desenvolver conceitos e estruturas mais complexos. Adicionalmente, a forma de definição de cores e a compreensão do referencial de coordenadas são outros 2 subtemas que constituem a base para um uso mais amplo desta linguagem visual.

Assim, depois de introduzidos estes elementos basilares, pode iniciar-se a introdução de outros conceitos, sejam estes específicos da linguagem Processing ou do domínio das linguagens de programação em geral (e.g., variáveis, instruções condicionais, ciclos de repetição). O facto de os resultados produzidos pelos programas terem um carácter visual é, desde logo, encorajador para os aprendizes. Esta opinião é também corroborada por pelos autores de (Soares et al., 2004), que refere a supremacia da eficácia na aprendizagem quando são usadas ferramentas visuais didáticas de representação de conceitos abstratos.

Resultado Sensorial como Objetivo

Habitualmente, na maioria dos contextos de aprendizagem de linguagens de programação, são propostos aos alunos problemas que se baseiam em resultados matemáticos obtidos a partir de fórmulas. Este facto só vem acrescentar ainda mais abstração e dificuldade à aprendizagem, pela natureza dos dados e das expressões numéricas que têm que ser usadas.

Não obstante a dificuldade envolvida, resta referir ainda a insipidez do resultado obtido: uma linha de texto com informação alfanumérica. Embora o aluno possa estar entusiasmado por ir conseguindo superar os objetivos dos exercícios que lhe são propostos, o que recebe como resultado é parco, comparativamente ao esforço que despendeu para o conseguir.

A metodologia aqui proposta passa por propor exercícios cujo resultado seja rico em elementos sensoriais que estimulem o aluno, jogando com a variedade em termos de cor, de forma, de movimento e de som. O facto de o resultado desafiar os sentidos do aprendiz é especialmente motivador para o seu empenho na aprendizagem da linguagem de programação, o que é um fator catalisador do êxito da aprendizagem. Tal revela-se importante, pois é sobejamente referido pelos autores de trabalhos neste domínio, nomeadamente em (Rodrigues, 2002), (Gomes et al., 2008) ou (Jenkins, 2002), que a falta de motivação é um entrave à aprendizagem desta matéria.

Também a tridimensionalidade, que é possibilitada pelo ambiente de desenvolvimento do Processing, é um outro aspeto que se destaca como atrativo para os alunos, pelo facto de as cenas criadas estabelecerem um paralelismo com o mundo real (3D) e, por isso, ser motivo de inspiração.

Do Simples para o Mais Complexo: Degrau a Degrau

É importante criar a sensação de confiança no aluno, para que este se sinta predisposto a querer alcançar algo mais. Por isso, no início de cada aula, deve-se começar por propor

um problema simples, que pressupõe pouco mais (ou nada mais do) que aquilo que o aluno já sabe fazer, com base nos conhecimentos já adquiridos. Segundo os autores de (Ausubel et al., 1980), o fator que mais influencia a aprendizagem é aquilo que o aprendiz já conhece.

Após a resolução problema, é relançado o mesmo problema, mas com um pouco de complexidade, de modo que o aluno possa aproveitar todo (ou quase todo) o código que já criou. Desta forma, o foco da sua atenção vai centra-se apenas no que há de novo neste problema e não no global.

O quadro seguinte contém o enunciado de um problema com múltiplas fases, ilustrativo da estratégia exposta.

Problema 1a. Crie uma sequência de 5 círculos concêntricos com raios sucessivamente menores em 10 unidades. Ao terminar o desenho do mais pequeno, o processo reinicia (automaticamente).

Problema 1b. Altere a alínea anterior, de modo a que cada sequência de 5 círculos concêntricos tenha localização aleatória.

Problema 1c. Altere a alínea anterior, de modo a que os círculos sejam sempre preenchidos com cores de saturação e brilho máximos (cores muito vivas e luminosas).

Se um novo degrau do problema implicar alguma novidade, para além do que o aluno já conhece, não deve ultrapassar mais que um elemento (e.g., um novo tipo de variável, uma nova instrução condicional, um novo ciclo de repetição, uma nova forma de definir a cor, etc.). Além disso, não deve ser seguida uma linha de sucessivas novidades, pois é fundamental a consolidação do que foi aprendido. Para isso, deve-se proceder à sistematização com exercícios/problemas de aplicação para consubstanciação do novo conhecimento adquirido (privilegiando sempre a diversidade de aplicação e não a vã repetibilidade).

Do Exemplo à Generalização

Tal como suprarreferido, um dos principais problemas da aprendizagem de programação reside na enorme abstração associada. Esta abstração ocorre a vários níveis, sendo disso exemplo as estruturas condicionais e os ciclos de repetição. A sua exposição como estrutura, descontextuada de um problema real que sirva para justificar a sua existência, subtrai-lhe a real importância. Além disso, a percepção sobre a sua usabilidade fica algo comprometida.

No entanto, ao promover condições de necessidade dessas estruturas, através de exercícios com problemas reais em que seja indispensável a sua existência, fomenta-se uma percepção mais consistente sobre o conceito.

Considere-se, por exemplo, a estrutura condicional “if”. Proponha-se, primeiro, um exercício em que uma linha horizontal se desloca de cima para baixo. Depois de resolvido, proponha-se que, quando essa linha chegar ao fundo da janela, passe a aparecer novamente ao cimo e, a partir daí, reinicie novamente o seu trajeto. Neste momento, impõe-se que seja facultado o conhecimento ao aluno sobre a estrutura condicional “if”,

de modo a poder dar solução ao problema. Contextualizado no problema, o “if” surge como a chave para descriptar o enigma.

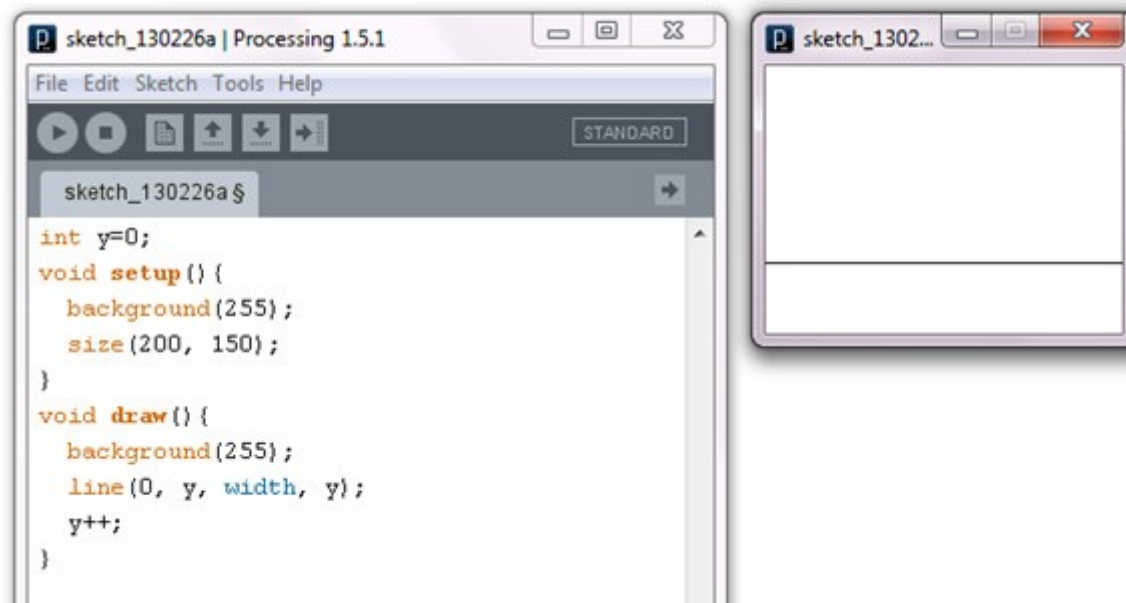


Figura 3 – Linha horizontal que se move no sentido descendente (código e screenshot do resultado)

O reforço imediato, com a apresentação de outros exercícios de aplicação em que seja usada a estrutura em causa, é imprescindível, como referido no ponto seguinte.

Além disso, é importante que, após a sistematização do uso da estrutura, através da realização de exercícios de aplicação, seja apresentada de uma forma formal, generalizando a sua forma. No caso de se adequar, devem ainda ser apresentadas outras considerações de carácter formal, como sejam fluxogramas ou diagramas.

O Reforço pela Sistematização

Numa primeira perspetiva, o professor pode ser levado a pensar que os alunos já dominam determinados conceitos, por terem conseguido resolver determinada problema. No entanto, na maior parte dos casos, esta perceção é ilusória. A consolidação dos conhecimentos só se obtém pela sistematização da sua aplicação, sendo conveniente variar na forma como estes aparecem em cada situação. Como são muitas as possibilidades de criação de cenas envolvendo efeitos gráficos (e sonoros), é possível conceber problemas de muito distintos, em que seja aplicado o conceito em causa.

De referir que esta sistematização não requer uma continuidade, existindo vantagem em revisitar a aplicação dos conceitos de forma mais alternada, ao longo do tempo.

Conclusões

A aprendizagem de linguagens de programação, essencialmente ao nível inicial, reveste-se de um conjunto de dificuldades, especialmente pela natureza da própria temática em si e pelo facto de implicar no aluno a habilidade de conciliar, simultaneamente, a sua capacidade de abstracção, o domínio da sintaxe e o talento para criar um algoritmo. Além

disso, as metodologias que são empregues no ensino de programação nem sempre são adequadas no sentido de apresentar os conteúdos de forma explícita, constituindo um entrave ao acessível entendimento dos aprendizes. Adicionalmente, leva à desmotivação, dando assim, um grande contributo ao insucesso da aprendizagem. Paralelamente, estes fatores levam a que a aprendizagem de programação seja uma atividade árdua e fastidiosa.

Tendo por base estas considerações, foram desenvolvidas estratégias metodológicas com o objetivo de contornar os principais procedimentos-padrão usados no ensino-aprendizagem de iniciação à programação. Primeiramente, a opção recai na escolha de uma linguagem de programação visual, que possibilita a criação de problemas baseados em imagem, o que é, de per si, uma mais-valia, pela envolvimento cativante. Além disso, são propostas metodologias que passam por explorar as características percetuais dos alunos, e que se relacionam com o raciocínio mais concreto e menos abstrato. Como tal, são omitidos os aspetos formais da linguagem (sintaxe e elementos estruturais), que são habitualmente inseridos como introdução, colocando-os numa fase última como conclusão. A generalização surge como resultado da operacionalização. O intuito consiste em focar sempre a atenção no problema e na sua resolução, relegando para segundo plano os elementos das questões de sintaxe, que surgem como uma ferramenta de suporte.

Desta forma, a aprendizagem de programação pode ser encarada pelos alunos de uma forma entusiasta, contrariando o habitual, com consequências claramente positivas para o êxito dos resultados.

Referências Bibliográficas

ALMEIDA, Eliana S. de [et al.]- AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação: X Workshop sobre Educação em Computação. Floianópolis: WEI 2002/SBC2002, 2002.

AUSUBEL, David; NOVAK, Joseph; HENESIAN, Helen - Psicologia Educacional. 2. 1980. ISBN 8520100848.

FONTES, Cléber R.; SILVA, Fabio W. O. da - O ensino da disciplina linguagens de programação em escolas técnicas. Ciências e Cognição. ISSN 1806-5821. Vol. 13, n.º 2 (2008), p. 84-98.

FRY, Ben - Visualizing Data. Paperback, 2007.

GLASSNER, Andrew S. - Processing for Visual Artists: How to Create Expressive Images and Interactive Art. Paperback, 2010.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António José - Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. Educação, Formação e Tecnologias. ISSN 1646-933X. Vol. 1, n.º 1 (2008), p. 93-103.

JENKINS, Tony- On the difficulty of learning to program: 3rd Annual LTSN_ICS. Loughborough: The Higher Education Academy, 2002.

REAS, Casey; FRY, Ben - Getting Started with Processing. O'Reilly Media/Make, 2010.

REAS, Casey; FRY, Ben - Processing: A Programming Handbook for Visual Designers and Artists. MIT Press, 2007.

RODRIGUES, M. C.- Como Ensinar Programação?: Boletim Informativo. ULBRA. Canoas, RS, Brasil, 2002.

SOARES, T. C. A. P. [et al.]- Uma Proposta Metodológica para o Aprendizado de Algoritmos em Grafos Via Animação Não-Intrusiva de Algoritmos: III Workshop de Educação em Computação e Informática do Estado de Minas Gerais (WEIMIG'2004). Belo Horizonte, MG, Brasil, 2004.

STAIR, Ralph; REYNOLDS, George - Principles of Information Systems. 6. Thomson Learning, Inc, 2003. ISBN 0-619-06489-7.